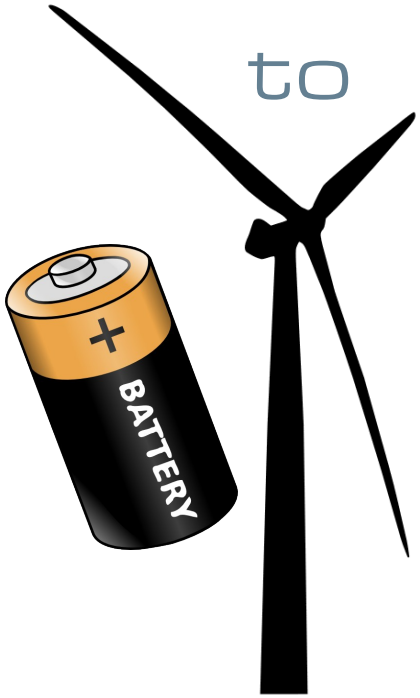
From Dynamic to Static

The challenge of depositing, archiving and publishing constantly changing content from the information environment

Dynamic to Static



?

Dynamic to Static

Archival objects

Stable identifiers

Simple and straightforward for end users

To integrate our publications management system with repositories

State of play... ?

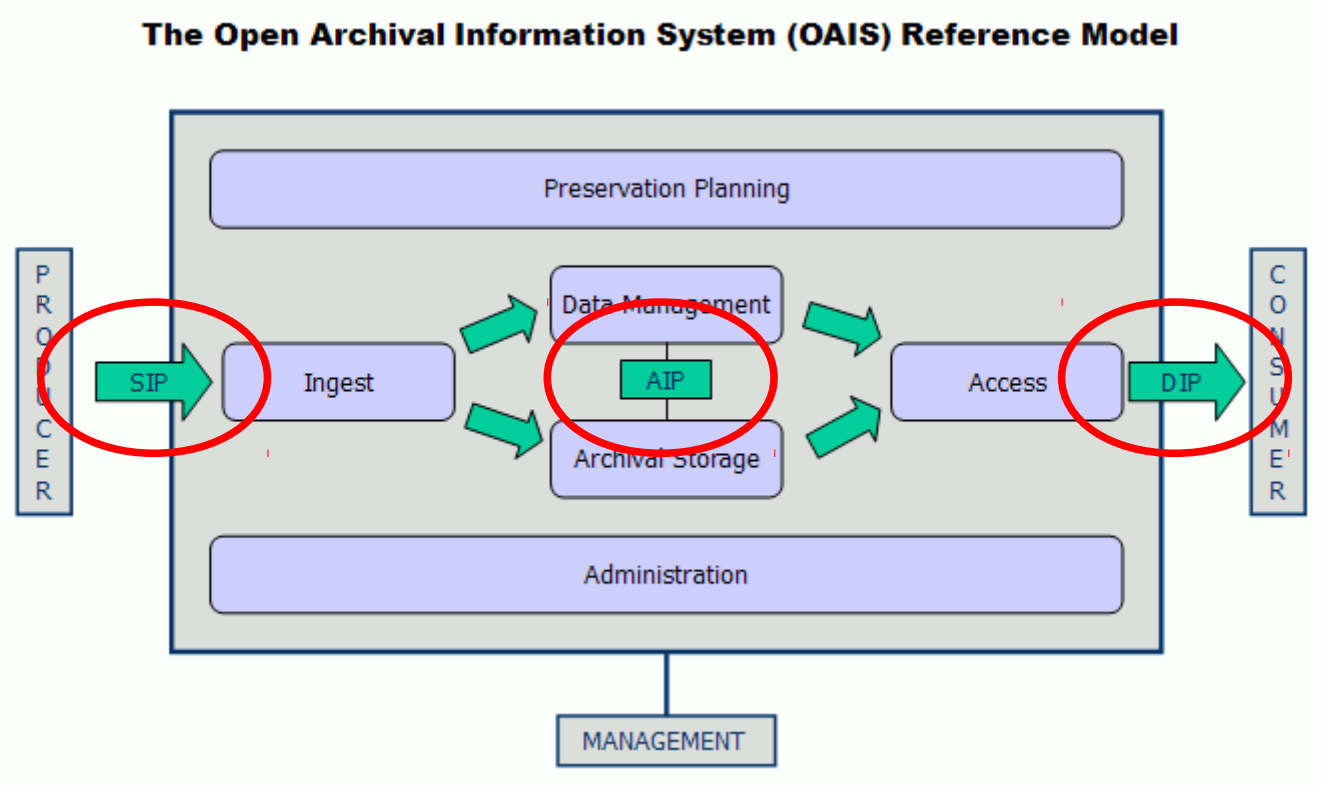
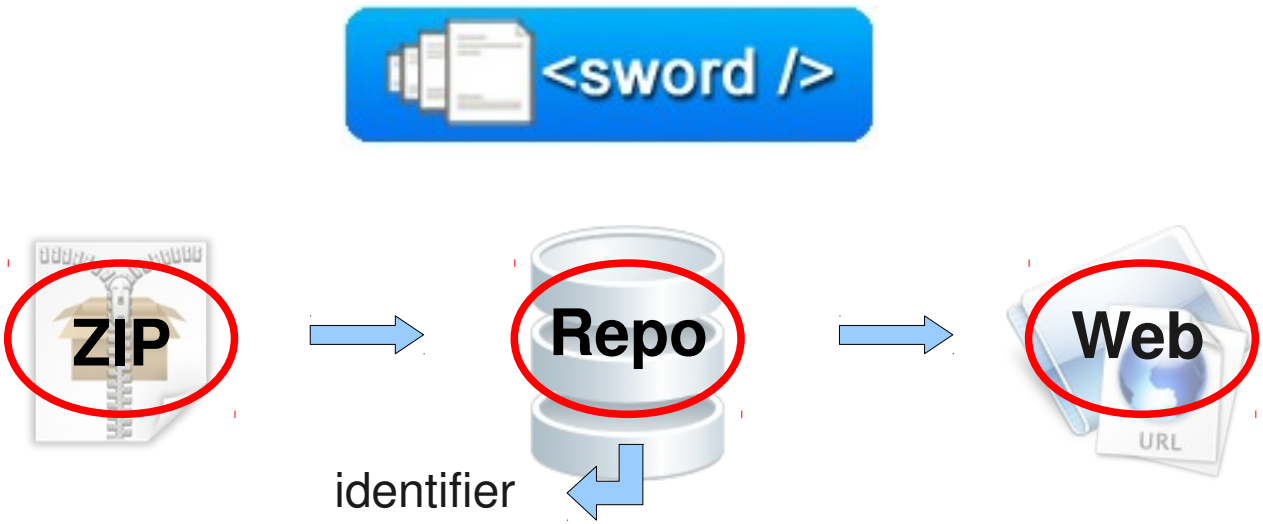


Image source: http://archivemati.ca/2005/11/08/digital-archives/

State of play... ?



Limitations to this approach...

It requires that content be stored outside the repository in an organised and well managed way prior to deposit; but this is what repositories are designed for!

It requires the end user to assert that a work is "finished"; this may not always be possible!

Solutions for this approach...

It requires that content be stored outside the repository in an organised and well managed way prior to deposit; but this is what repositories are designed for!

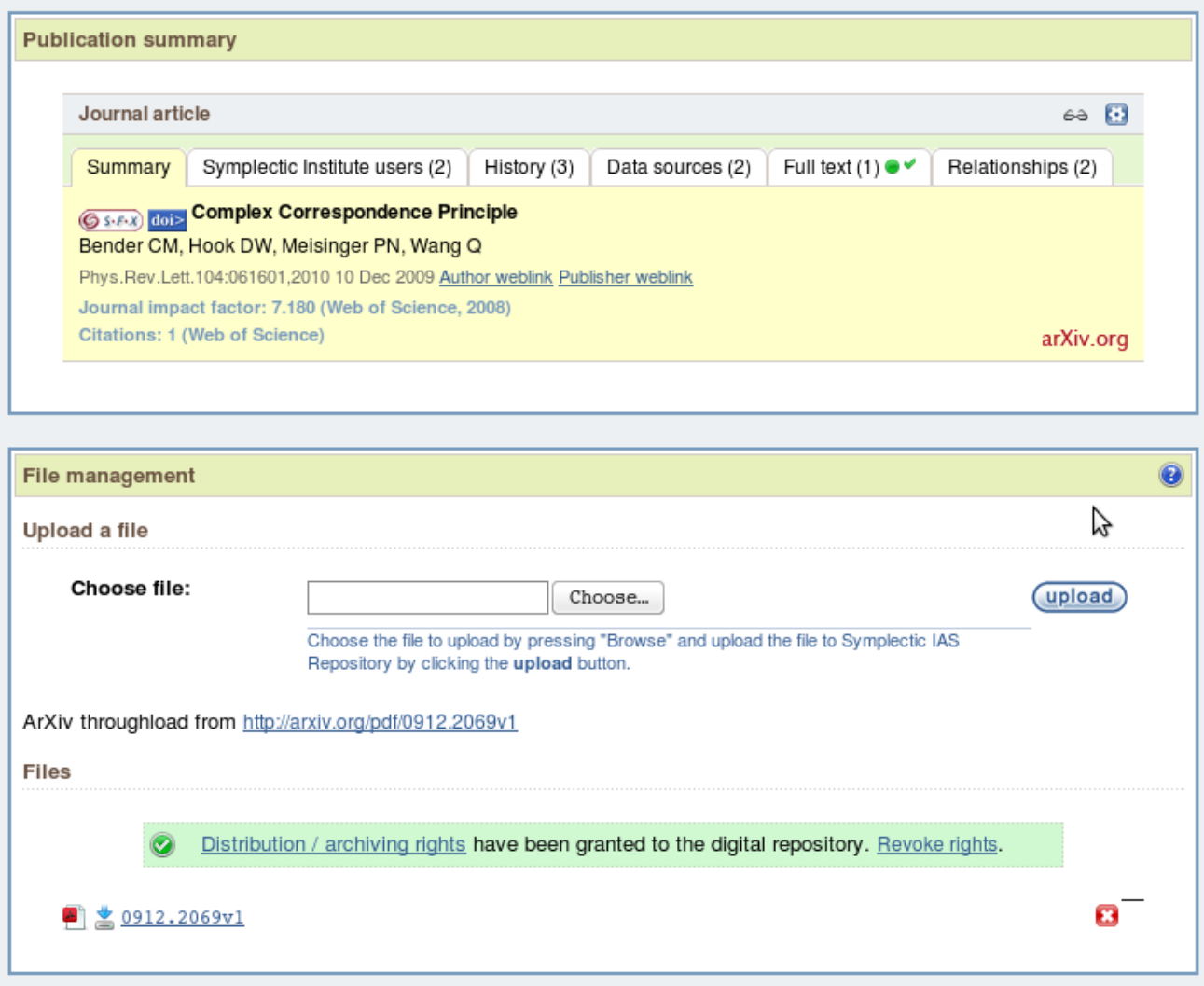
*CRUD - **C**reate **R**etrieve **U**pdate **D**elete*

It requires the end user to assert that a work is "finished"; this may not always be possible!

Workflow

Versioning

Solution ... Symplectic Repository Tools



Solution ... Symplectic Repository Tools

The repository is the authority for the full-text

The repository manager is free to configure the repository workflow as normal

The end user does not HAVE to think about the repository

Symplectic Repository Tools

Academic Use Cases

1. pre-print archived
2. peer reviewed replaces it
3. published replaces that
4. source data added

Metadata automatically acquired

Repository Use Cases

1. pre-print published
2. peer reviewed published
3. file formats converted
4. mediated deposit

Variables

- different article versions
- supporting documentation
- alternative versions/formats/qualities
- source data

Create Retrieve Update Delete

AtomPub + SWORD = Full CRUD +
Open Standards +
Appropriate for repositories

Implementations for DSpace, EPrints, Fedora and IntraLibrary...

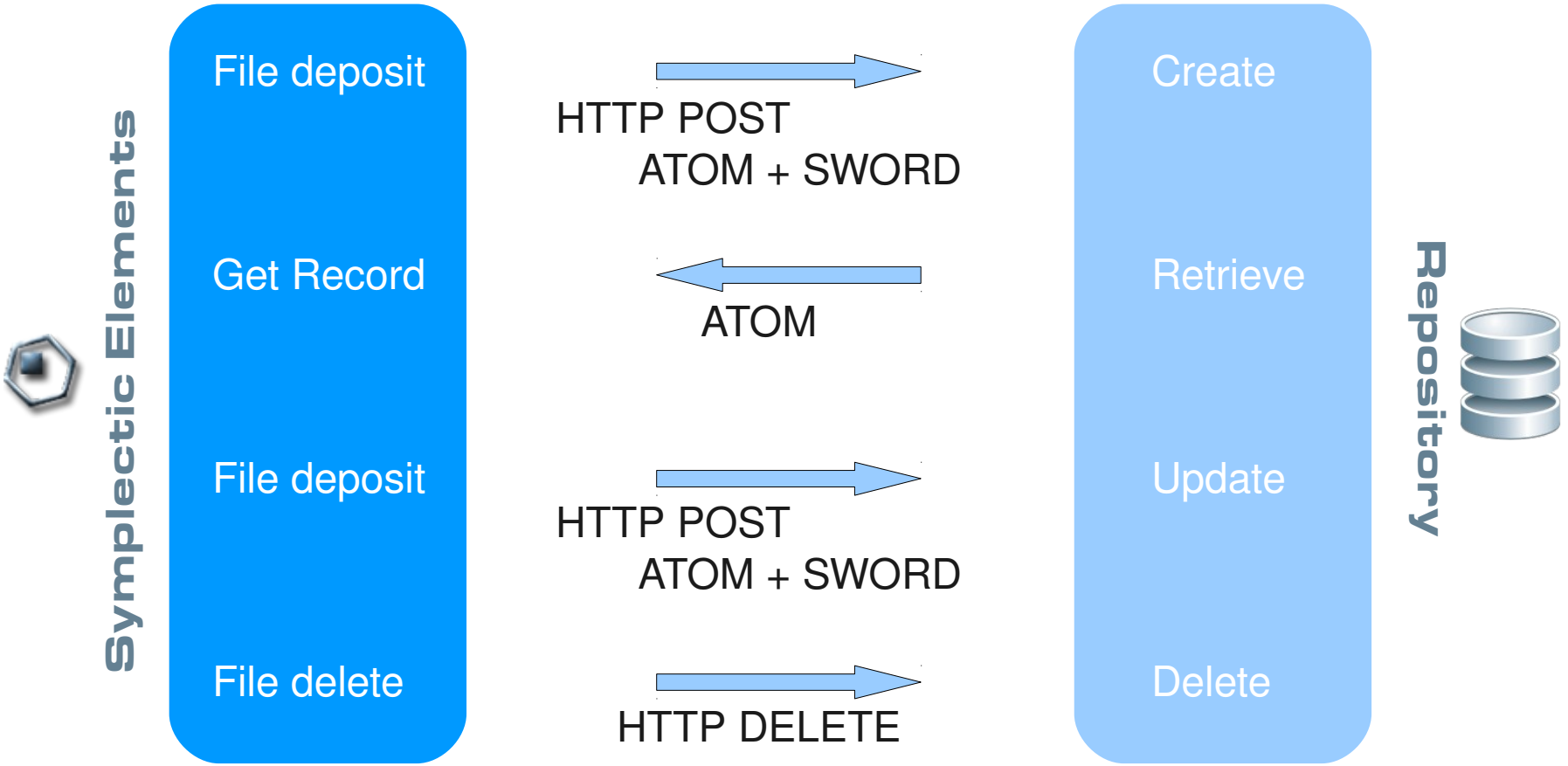
Data Models and ATOM

ATOM	DSpace	EPrints	Fedora
Collection	Community/Collection	Dataset	-
Feed	Item	EPrint	Object
-	Bundle	Document	-
Entry	Bitstream	File	Datastream

symplectic

solutions for higher education

CRUD with AtomPub + SWORD



Workflow Integration

	In Progress	In Review	Archive
DSpace	Workspace	Workflow	Archive
EPrints	Inbox	Buffer	Archive
Fedora	[Symplectic Lock]	[Oxford Lock]	[Visible]

Utilised workflow; available in UI where appropriate

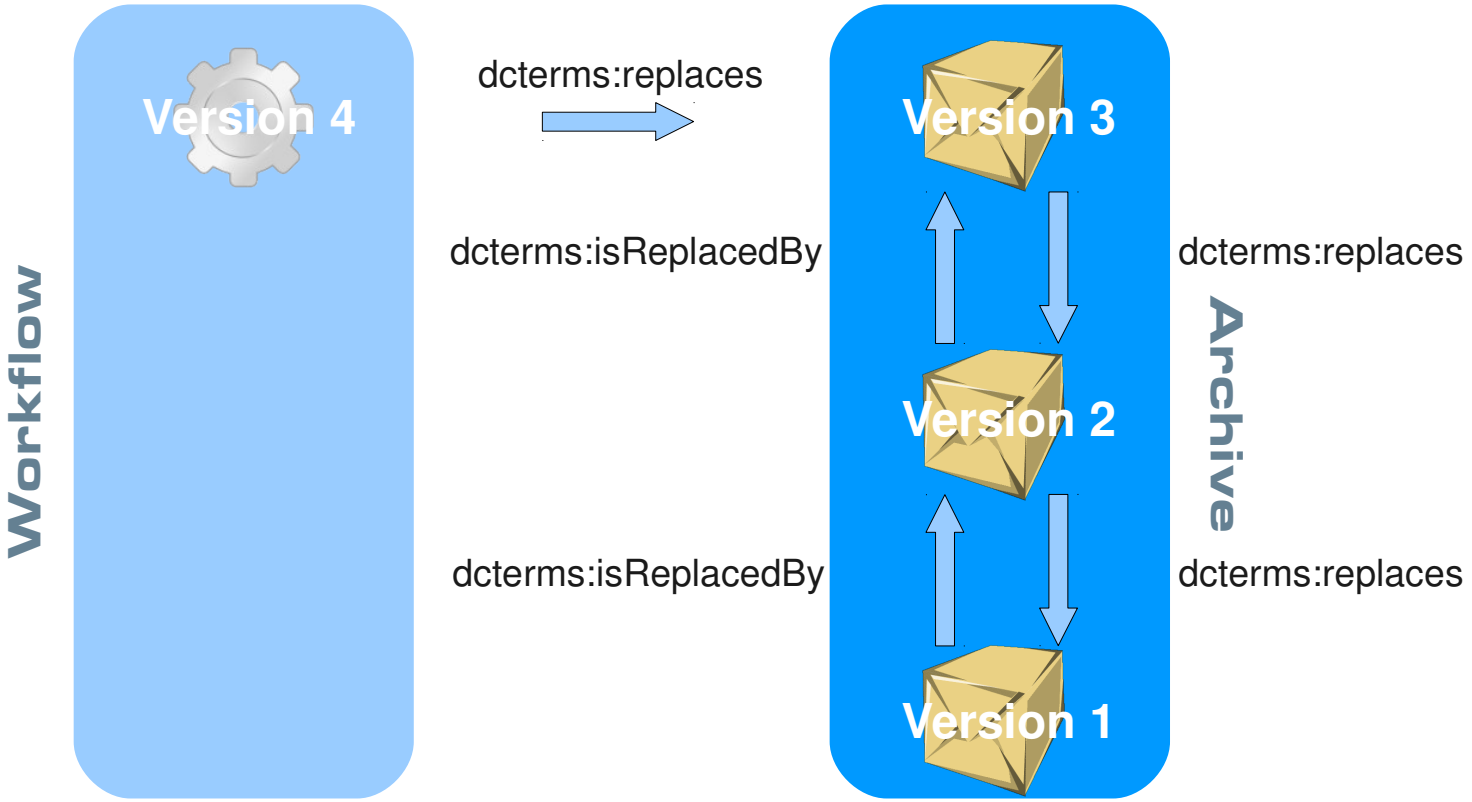
Additional API workflow terms; not utilised

Versioning

Versioning Type	Technical	Relational	Multiple parent/child	UI Support
DSpace	✗	✓	✓	✗
EPrints	✗	✓	✗	✓
Fedora	✓	✓	✓	✗

- **DSpace**: All done by metadata fields
- **EPrints**: succeeds/replaced by are singletons in the eprint metadata, hence no multiple parent/child relationships
- **Fedora**: Technical versioning is built in, but other forms of versioning have to be applied by implemeter, which is relatively easy

Versioning (2)



What have we achieved here?

- We can place appropriately structured content into 3 different repositories using an open standard
- The API is CRUD, so all operations required to connect the repository to another working environment are provided
- The repository workflow is uninhibited by the deposit API, but the all deposits are integrated smoothly into it
- The Repository Manager chooses when to commit to the archive, the depositor can just file-manage
- Archiving does not interrupt the deposit process; version chains are maintained

Walkthrough

[see video]

Conclusions

- Seamless repository integration with Symplectic Elements (a research management system)
- Uses open standards (HTTP, AtomPub, SWORD)
- Feeding back the results of our work to SWORD
- Reduces barriers to deposit, increasing throughput to the repository
- More accurately reflects requirements for the constantly changing state of the information environment

Richard Jones

richard@symplectic.co.uk

http://www.symplectic.co.uk/products/repository-tools.html